Dynamic class-balanced threshold federated semi-supervised learning by exploring diffusion model and all unlabeled data

Zeyuan Wang, Yang Liu, Guirong Liang, Cheng Zhong, Feng Yang

PII: S0167-739X(25)00115-3
DOI: https://doi.org/10.1016/j.future.2025.107820
Reference: FUTURE 107820

To appear in: *Future Generation Computer Systems*

Received date : 24 December 2024
Revised date : 5 March 2025
Accepted date : 13 March 2025

Click here to view linked References

Highlights

Dynamic Class-Balanced Threshold Federated Semi-Supervised Learning by Exploring Diffusion Model and All Unlabeled Data

Zeyuan Wang,Yang Liu,Guirong Liang,Cheng Zhong,Feng Yang

- Innovative Federated Semi-Supervised Learning approach using diffusion model and all unlabeled data
- Generating datasets that satisfy global distributions with diffusion model
- Dynamic Class Balancing Thresholds adaptively adjust thresholds for each class and distinguish high and low confidence samples
- Residual Class Negative Learning using low confidence samples for model optimization
- Outperforms all previous methods on two public datasets

Dynamic Class-Balanced Threshold Federated Semi-Supervised Learning by Exploring Diffusion Model and All Unlabeled Data

Zeyuan Wang[a], Yang Liu[b], Guirong Liang[a], Cheng Zhong[a] and Feng Yang[a,c,*]

[a] *School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, 530004, PR China*
[b] *Center for Machine Vision and Signal Analysis, University of Oulu, Oulu, FI-90014, Finland*
[c] *Guangxi Key Laboratory of Multimedia Communications Network Technology, Guangxi University, Nanning, Guangxi, 530004, PR China*

ARTICLE INFO

Keywords:
Federated Semi-Supervised Learning
Diffusion Model
non-IID
Image Classification
Data Augmentation
Class-Balanced Threshold

ABSTRACT

Federated Semi-Supervised Learning (FSSL) aims to train models based on federated learning using a small amount of labeled data and a large amount of unlabeled data. The limited labeled data and the issue of non-independent and identically distributed (non-IID) data are the major challenges faced by FSSL. Most of the previous methods use traditional fixed thresholds to filter out high-confidence samples and assign pseudo-labels to them without considering low-confidence samples. These methods then increase the sample space by random sampling and other techniques to address the challenges of FSSL. However, the performance of these models remains unsatisfactory. To tackle these challenges, we propose DDRFed, a novel FSSL framework that effectively utilizes all available data by integrating a diffusion model and dynamic class balance thresholds. Specifically, we first mitigate the client-side non-IID issue by utilizing a dataset generated by a client-side co-trained diffusion model that conforms to the global data distribution. The local clients then use the global class distribution information provided by the server to establish dynamic class balance thresholds, which distinguish between high-confidence and low-confidence samples. The existence of dynamic thresholds ensures a sufficient amount of labeled data during the training process. Meanwhile, to fully leverage the knowledge contained in low-confidence samples, we optimize the model's performance through residual class negative learning. Experiments conducted on two natural datasets demonstrate the superiority of DDRFed, addressing both major challenges in FSSL.

1. Introduction

With the rapid development of fields such as Artificial Intelligence (AI) and the Internet of Things (IoT), the number of benefiting users is steadily increasing, and conventional computing platforms are gradually transitioning to various mobile applications in daily life.However, numerous privacy and security risks have also emerged, leading to the problem of "data silos." To address this issue, McMahan et al.[1] proposed Federated Learning and the FedAvg algorithm in 2017. The goal was to use a distributed machine learning framework to train local models on decentralized and localized datasets, thereby tackling the data privacy and security challenges inherent in traditional centralized learning methods. In the Federated Learning paradigm, each client only needs to upload its locally trained model to the server. The server aggregates the collected models and sends the updated model back to the clients. This approach enables training through model exchange among clients without exposing local private data.

Although FedAvg has achieved promising results, in real-world scenarios, data annotation often requires specialized expertise and substantial resources. As a result, research in Federated Learning has gradually shifted towards Federated Semi-Supervised Learning (FSSL), which has become a widely explored topic among researchers[2][3][4][5]. Federated Semi-Supervised Learning can be broadly categorized into two types based on the application scenario. The first type assumes that all labeled data resides on the server, while the clients lack any labeled data[2]. The second type assumes that labeled data is distributed among the clients, which can be further divided into two cases: (1) each client possesses a small portion of labeled data[3], or (2) a subset of clients has all the labeled data (labeled clients), while the other clients hold only unlabeled data (unlabeled clients)[4][5]. This paper primarily focuses on the latter case of the second type.

In Federated Semi-Supervised Learning (FSSL), there are two main challenges. The first challenge is the limited availability of labeled data, which can lead to training bias in clients with unlabeled data. Traditional threshold-based pseudo-labeling methods utilize only a small portion of the data in each training round, leaving a significant amount of data underutilized. The second challenge arises from the heterogeneity of local data distributions among users in real-world scenarios. This non-independent and identically distributed (non-IID) nature of the data is one of the reasons for poor model performance in FSSL training. Joeng et al.[2] proposed the FedMatch algorithm, which aims to enhance model generalization by promoting consistency among different clients. However, it does not extensively address the issue of heterogeneous data. FedIRM[6] introduced a relationship-matching approach among clients to improve model performance in medical image classification. However, it is based on the assumption of homogeneous

*Corresponding author
wzy625@foxmail.com (Z. Wang); yang.liu@oulu.fi (Y. Liu); grliang@foxmail.com (G. Liang); chzhong@gxu.edu.cn (C. Zhong); yf@gxu.edu.cn (F. Yang)
ORCID(s): 0000-0002-1898-3784 (F. Yang)

client distributions, making it suboptimal in non-IID settings. Consequently, the lack of labeled data and the non-IID nature of data are critical obstacles for effective model training.

Therefore, we propose DDRFed to fundamentally address the challenges in FSSL, where the scarcity of labeled data limits the effective utilization of most data and the non-IID problem. Specifically, inspired by diffusion models[7][8], this study introduces a globally distributed data generation module (GDGM) based on diffusion models. First, a diffusion model is trained using federated learning, enabling the server to generate a dataset that adheres to the global distribution. This dataset is then randomly and evenly distributed to clients, effectively mitigating the non-IID problem without exposing the original local data of the clients. Secondly, we construct a class-balanced dynamic threshold (DCBT) using the difference between the global class distribution and its standard deviation, along with the double exponential moving average (DEMA). It dynamically adjusts thresholds for different classes based on the model's learning status to ensure class balance. By integrating the diffusion model with DCBT, we can fundamentally resolve the non-IID problem in FSSL.Finally, to fully utilize all unlabeled data and mitigate training bias, we introduce Residual Class Negative Learning (RNL) on the client side. RNL leverages consistency optimization between two different augmented versions of the data to estimate the accuracy of low-confidence samples. Confidence suppression is then applied to residual classes, refining overall model performance and strengthening class prediction capabilities. Overall, our contributions can be summarized as follows:

- We propose a novel Federated Semi-Supervised Learning method called DDRFed. This method introduces a Global Distribution Data Generation Module (GDGM) based on a diffusion model. After local client training and server aggregation, the GDGM generates a globally distributed synthetic dataset. The server then distributes this dataset to each client in a manner consistent with the global distribution, effectively addressing the non-IID problem in Federated Learning.

- We propose a novel Dynamic Class-Balanced Threshold (DCBT) method, which is constructed based on the global data distribution and standard deviation of the global data distribution in each federated communication round, combined with Double Exponential Moving Average (DEMA). This method adaptively adjusts the threshold for each class to suit different learning stages, thereby fully utilizing high-confidence samples and produce accurate pseudo-labels.

- We propose a Residual Class Negative Learning (RNL) method, which evaluates the $top-n$ performance of low-confidence samples by leveraging the consistency between two different augmentations of the data. Negative labels are assigned to residual classes to suppress their confidence, thus making full use of the unlabeled data, enhancing the learning ability of the model and improving its overall performance.

- Experiments conducted on two public datasets, CIFAR-10 and Fashion-MNIST, demonstrate that our DDRFed outperforms state-of-the-art FSSL methods, achieving a 6.93% accuracy improvement on CIFAR-10 and a 2.17% accuracy improvement on Fashion-MNIST.

2. Related work

2.1. Federated learning

Federated Learning (FL), with its unique multi-machine collaborative training approach and emphasis on privacy protection, has become a widely recognized distributed machine learning paradigm in recent years. However, FL faces various challenges, such as data and device heterogeneity[9], privacy protection[10], and communication efficiency[11]. In most FL scenarios, the distributional differences in data held by different clients have made data heterogeneity a persistent challenge in FL. FedAvg[1] is one of the most widely used algorithms, but it is only suitable for non-heterogeneous or simple heterogeneous scenarios. Methods to address data heterogeneity can be broadly categorized into local client training approaches[12][13] and global model aggregation methods[14][15]. MOON[12] alleviates the non-IID problem by introducing contrastive learning, which reduces the feature representation gap between local and global models, while also increasing the distance between the local model and the previous round's model representation. SCAFFOLD[13] posits that non-IID data causes "client drift" in local clients. It uses control variables (variance reduction) to estimate the update directions of clients and the server, correcting the "client drift" in local updates. FedProx[14] reduces the difference between local and global models by adding an $L2$ regularization term to the loss function. FedBE[15] performs robust aggregation by sampling higher-quality global models and combining them using a Bayesian model, transforming them through Bayesian inference.

2.2. Semi-supervised learning

Semi-supervised learning combines a small amount of labeled data with a large amount of unlabeled data for training, aiming to leverage the latent information in the unlabeled data to improve model performance. Currently, mainstream semi-supervised methods often combine consistency-based approaches and pseudo-labeling techniques to achieve better results[16][17][18][19]. Additionally, the quality of the generated pseudo-labels and the final performance of the model are closely related to the threshold selection. FixMatch[16] uses a fixed high threshold to filter all classes. The drawback of this approach is that the model does not adequately account for the varying training conditions and difficulty of training across different classes, leading to poor performance in class-imbalanced scenarios. FlexMatch[17] introduces a curriculum pseudo-labeling method that dynamically adjusts the threshold for each class throughout the training process, without introducing additional parameters or computational overhead. FreeMatch[18] adaptively adjusts the confidence threshold based on the model's learning state, while introducing an adaptive class fairness regularization penalty to encourage diverse predictions and improve performance in imbalanced SSL settings. SoftMatch[19] still uses dynamic thresholds, focusing on balancing the number and quality of pseudo-labels. It demonstrates the importance of maintaining pseudo-label quality while maximizing the utilization of unlabeled data in SSL.

2.3. Federated semi-supervised learning

Federated Semi-Supervised Learning (FSSL) generally refers to semi-supervised classification tasks conducted within the federated learning paradigm, utilizing a small amount of labeled data and a large amount of unlabeled data. In the first type of FSSL scenario mentioned earlier, FedMatch[2] trains by decomposing parameters between different types of data and introduces consistency loss between clients to ensure that effective information contained in unlabeled data is learned from each sample. imFed-Semi[20] proposes a dynamic bank learning scheme, where labeled data from the server guides the construction of dynamic banks to extract category proportion information for each local client, converting the original classification task into a sub-bank-based classification task, thus improving client training. In the second type of FSSL scenario, FedDure[21] and FedSSL[22] discuss the situation of jointly training labeled and unlabeled data within clients. FedDure[21] introduces a dual-regulator mechanism, applying different regulators to the client and global model aggregation processes to balance the model's learning process, thus alleviating the issue of data heterogeneity among clients in FSSL. FedSSL[22] designs a data mixing augmentation method based on a generative model to leverage distributed data sources, allowing the model to generate data from a single feature space to mitigate heterogeneity. Fed-Consist[23], RSCFed[4], CBAFed[5], and PDCFed[24] discuss scenarios involving training fully labeled clients and unlabeled clients. Fed-Consist[23] combines consistency regularization and pseudo-labeling methods in FSSL but does not account for the heterogeneity of data distributions both between and within clients. RSCFed[4] does not directly aggregate the models but first performs random sampling on the clients to extract multiple sub-models, which are then aggregated into the global model. It also proposes a new distance-weighted model aggregation method (DMA) to dynamically fine-tune the weights of each local client's sub-model. CBAFed[5] designs a class-balanced adaptive threshold by considering the distribution of all training data in local clients to encourage balanced training and introduces a residual weight connection method to achieve better performance. PDCFed[24] performs two-stage sampling by predicting distribution changes under different data augmentations and assesses sample credibility based on the maximum predicted probability of weak augmentations. Samples in unreliable regions are further filtered after being adjusted by a Gaussian function, helping to alleviate the non-IID problem.

Most of the above FSSL methods use traditional fixed thresholds to filter samples. When labeled data is limited, excessively high thresholds result in most data being discarded in the early stages, leaving only a small portion of data with pseudo-labels for model training. This is not conducive to the model's early learning. Moreover, due to the impact of thresholds, the handling of the non-IID problem in these methods is also affected, and they do not fundamentally resolve the two major challenges of FSSL. Our DDRFed first assigns a globally distributed dataset generated by GDGM to the clients to alleviate the non-IID problem. Then, it uses DCBT to separately select high-confidence and low-confidence samples for the clients. High-confidence samples are trained normally, while low-confidence data is handled using RNL to enhance the model's learning ability and performance. In summary, our method uses GDGM to address the non-IID problem in FSSL, while DCBT and RNL tackle the issue of limited labeled data and the inability to fully utilize all data.

2.4. Diffusion model

Diffusion models are a class of generative models based on the principles of non-equilibrium thermodynamics, which have made significant progress in generative tasks in recent years. They simulate the gradual diffusion process of data distributions, first converting real data into Gaussian noise, and then performing a reverse denoising process to generate new samples. In [25], diffusion models are categorized into three forms. First, Denoising Diffusion Probabilistic Models (DDPMs) [7][26] estimate the probability distribution of image data by applying the diffusion process at discrete time intervals. Both the forward and reverse processes can be viewed as Markov chains. DDPMs are currently the most widely studied, with stable training and high-quality image generation. Next, Noise Conditional Score Networks (NCSNs) [27] train a shared neural network using score matching to estimate the score function of disturbed data distributions at different noise levels. However, their training is unstable, resulting in poor details and realism in the generated images. Additionally, they are highly dependent on the sampling strategy, and the generated quality may be inconsistent across different noise levels. Finally, Stochastic Differential Equations (SDEs) [28] represent an alternative approach to simulating diffusion. By modeling diffusion with forward and reverse SDEs, effective generative strategies can be obtained. However, SDEs are limited by the accuracy of the solvers, and numerical errors may occur during reverse sampling, which affects the generation quality. Additionally, their generation speed is slower, and the sampling cost is higher. Therefore, this paper focuses on DDPMs [7][26], which are currently widely studied, have stable training, high-quality image generation, and strong diversity.

Meanwhile, diffusion models have also shown a wide range of applications in the field of SSL and FSSL, [29]proposed a double pseudo-training approach, where a classifier is first trained on partially labelled data to generate pseudo-labels, followed by using the pseudo-labels to train a conditional diffusion model to generate pseudo-images, and finally re-training the classifier by combining the real images and pseudo-images. [30] introduced diffusion models into FSSL by using pre-trained diffusion models by the server to generate synthetic datasets that match the client's distribution and train a global model on them, with performance reaching even the upper limit of SSL training.

3. Methodology

In Section 3, we first provide an explanation of the FSSL framework involved in this paper and define the parameters, then describe our motivation and provide a detailed explanation of DDRFed.

3.1. Definitions and Framework

In our FSSL problem setting, we consider the training of fully labeled clients and fully unlabeled clients. Suppose there are n labeled clients $\{C_1, \dots, C_n\}$ and m unlabeled clients $\{C_{n+1}, \dots, C_{n+m}\}$(generally, $n \geq 1, m \gg n$). The local dataset of the labeled client C_l is denoted as $D_l = \{X_i^l, Y_i^l\}_{i=1}^{N_l}$, $l = 1, 2, \dots, n$,where N_l represents the number of samples in D_l. Similarly, the local dataset of the unlabeled client C_u is denoted as $D_u = \{X_i^u\}_{i=1}^{N_u}$, $u = n+1, n+2, \dots, n+m$,where N_u represents the number of samples in D_u. Our goal is to train a global model θ_g using the local data of clients and the generated dataset from the diffusion model, such that it has strong classification ability and robustness.

Fig.1 illustrates our FSSL framework. The training procedure is divided into two parts: the globally distributed data generation phase based on the diffusion model (steps ① and ② in the figure), and the federated semi-supervised learning phase (steps ③ and ④ in the figure). The method is as follows:

(1) Globally Distributed Data Generation Phase based on the Diffusion Model(preheating phase): ① Each client trains a diffusion model multiple times using its local dataset and uploads the trained local model to the server. The server aggregates the diffusion models from all clients until the specified number of communication rounds is reached. ② The server uses the final aggregated diffusion model to generate a new dataset D_g with a global feature distribution, which is then evenly split into $\{D_g^1, ..., D_g^{n+m}\}$. This split dataset $\{D_g^1, ..., D_g^{n+m}\}$ is distributed to all clients, where each client merges its local dataset with the corresponding $\{D_g^1, ..., D_g^{n+m}\}$ (GDGM).

(2) Federated Semi-Supervised Learning Phase: ③ Clients dynamically update the threshold for each class (DCBT) based on the model's state and the difference in global class distribution and standard deviation provided by the server. DCBT is used to differentiate between high-confidence and low-confidence samples. High-confidence samples are used for training, and the local class distribution is recorded. For low-confidence data, the consistency between different augmented versions is optimized to compute $top-n$ accuracy, and negative pseudo label constraints are applied to the classes outside the $top - n$ (residual classes) to fully utilize all samples (RNL). After training, the client uploads the model to the server. ④ The server collects and aggregates the models trained by clients, computes the global class distribution, and sends the aggregated model and global class distribution to each client. Steps ③ and ④ are repeated until the specified number of communication rounds is reached.

3.2. Global Distribution Data Generation Module Based on Diffusion Models

3.2.1. *Federated Diffusion Optimization Objective*

Diffusion models, as a type of generative model, generate images matching the target distribution through two steps: forward noise addition and reverse denoising. Given a sample $x_0 \sim q(x)$,in the forward noise addition process shown in Eq.(1) and (2), Gaussian noise is added iteratively over T steps,resulting in a series of progressively noisier samples $x_1, x_2, \dots, x_T$. The magnitude of noise at each step is controlled by a set of Gaussian variance hyperparameters $\{\beta_t \in (0,1)\}_{t=1}^{T}$. Since each state t in the forward process depends only on $t-1$,the process can be viewed as a Markov process, as shown in Eq.(1) and (2).

$$q(x_{1:T}|x_{t-1}) = \prod_{t=1}^{T} q(x_t|x_{t-1}), \tag{1}$$

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, \beta_t I), \tag{2}$$

In the above equations, $\sqrt{\alpha_t}$ is the mean coefficient, and $\alpha_t = 1 - \beta_t$.It can be observed that the forward process possesses a unique property, allowing x_t to be directly obtained from x_0 and a noise sample β.As shown in Eq.(3),where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$.

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I), \tag{3}$$

The reverse denoising process reverses the above procedure by sampling from $q(x_{t-1}|x_t)$, and reconstructs the original image distribution $x_0 \sim q(x)$ from Gaussian noise $x_T \sim \mathcal{N}(0, I)$. Therefore, a neural network ω is used to predict this reverse distribution p_ω, as shown in Eq.(4) and (5).

$$p_\omega(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\omega(x_{t-1}|x_t), \tag{4}$$

$$p_\omega(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\omega(x_t, t), \sum\nolimits_\omega (x_t, t)), \tag{5}$$

We can now confirm that in the forward process, the mean and variance of each conditional probability depend on the known β_t and x_0, whereas in the reverse process, the mean and variance need to be learned by the neural network. Therefore,$\mu_\omega(x_t, t)$ is reparameterized as a function of $\epsilon_\theta(x_t, t)$, which represents the noise ϵ_t to be subtracted from sample x_t. Equation (6) shows the reparameterization

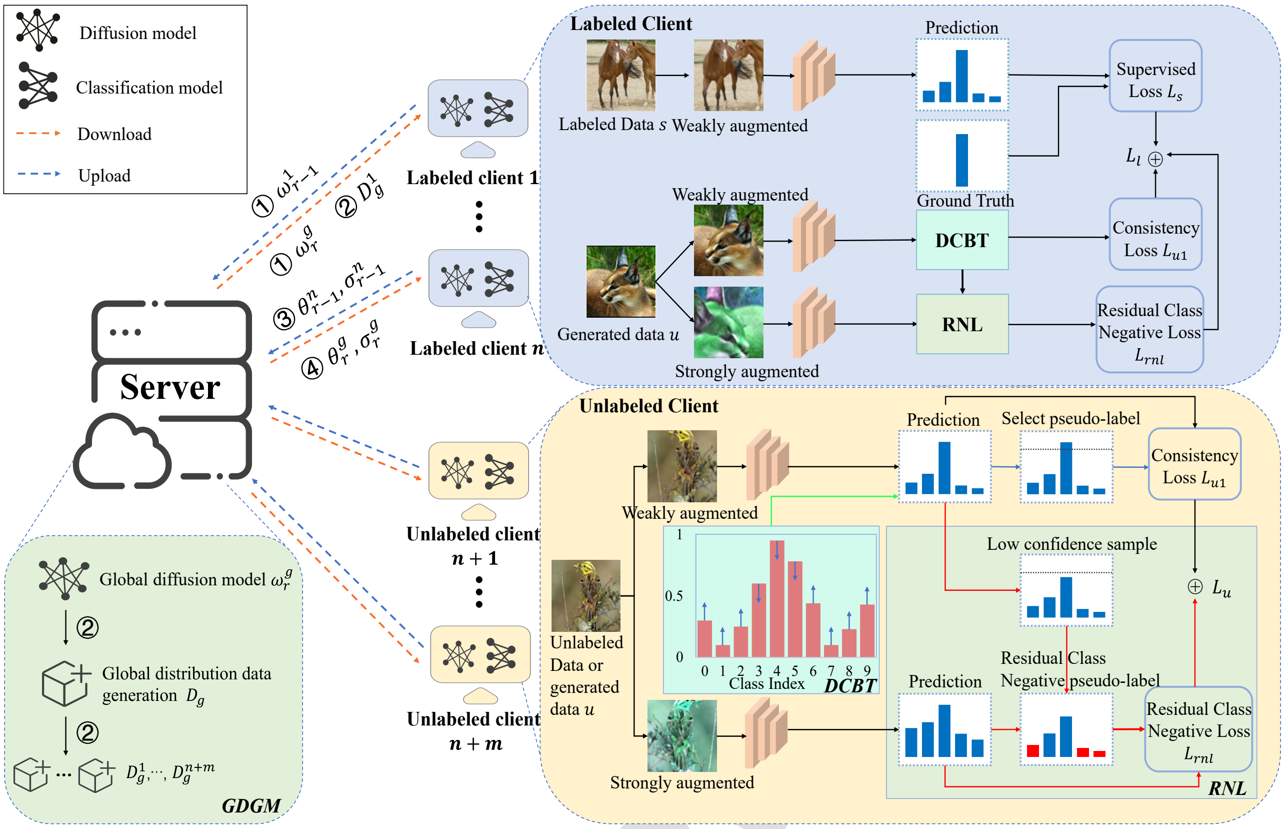


Fig. 1: The overall framework of our proposed DDRFed is divided into a globally distributed data generation module based on diffusion models (warm-up phase) and a federation training phase, where the server utilizes the trained diffusion models to generate new datasets for clients to use in the second phase. In the federated learning phase, all clients train the model using the class balance dynamic threshold and the residual class negative loss. The red line indicates residual class negative learning, the green line indicates dynamic class balancing thresholds, and the global distribution data generation module is at the client.

process of $\mu_\omega(x_t, t)$.

$$\mu_\omega(x_t, t) = \frac{1}{\sqrt{1-\beta_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\omega(x_t, t)), \tag{6}$$

Combining Equation (5), we can derive the formula for sampling x_t in one step, as shown in Eq.(7).

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, \tag{7}$$

Therefore, the optimization objective of the diffusion model becomes minimizing the distance between the noise ϵ_t and the model's noise estimate $\epsilon_\omega(x_t, t)$ for each time step t.

$$L_{dm}(\omega) = \mathbb{E}_{t,x,\epsilon_t\sim\mathcal{N}(0,I)} \parallel \epsilon_t - \epsilon_\omega(x_t, t) \parallel_2^2, \tag{8}$$

In the FSSL scenario, using only a small amount of labeled data and conditional diffusion models leads to poor quality of generated samples. Therefore, we use an unconditional diffusion model for sample generation. Algorithm 1 represents the pretraining phase in our FSSL framework. Below, we provide a detailed explanation of how diffusion models are trained in the FL setting. In the pretraining phase of our FSSL framework, each client maintains a diffusion model. Initially, client-side local data follows a strictly non-IID partitioning strategy. During each training round, all clients update their local models in parallel. After several rounds of local training, the stabilized model parameters are uploaded to the server. The server then applies the FedAvg[1] algorithm to aggregate the model parameters received from clients. Given the characteristics of diffusion models, EMA smoothing is applied post-aggregation to reduce model drift in training. To reduce communication overhead, we adopt a fixed-round synchronization strategy, where training is considered complete once a predefined number of client-server interactions have occurred. The ultimate optimization objective of training diffusion models using FedAvg [1] is as follows:

$$L(\omega) = \sum_{k=1}^{n+m} \frac{|D_k|}{|D|} F_k(\omega), where F_k(\omega) = L_{dm}(\omega). \tag{9}$$

Here,D_k represents the local dataset of client C_k, $|D|$ denotes the total data size, and $F_k(\omega)$ indicates the local loss of client C_k.

After training is completed, the server uses the pretrained diffusion model to generate a synthetic dataset D_g that matches the global data distribution. The size of dataset

D_g is determined by the generation coefficient ψ multiplied by the total size of all clients' local datasets. This dataset is then randomly allocated to clients in equal amounts. Due to the nature of the unconditional diffusion model, we can assume that the distribution of the allocated data is approximately iid. Subsequently, clients incorporate this data into the second phase of training to alleviate the non-IID data issue, while ensuring the privacy of the clients' local data.

3.2.2. *Federated Diffusion Unet Network*

In the federated diffusion learning process, for each client, we consider using the classic Unet network, which adopts an encoder-decoder structure and retains high-resolution information through skip connections. The encoder extracts low-level features of the image through downsampling operations, reducing the spatial resolution of the image while increasing its dimensions. The task of the decoder is to recover the spatial resolution of the image through upsampling operations, and to merge the feature maps of the encoder with those of the decoder via skip connections, thereby preserving effective information. However, federated learning is affected by the heterogeneity of data partitioning, which may lead to poor model performance for certain clients. Inspired by ADA[31], we apply the Adaptive Augmentation pipeline not only in the early data preprocessing phase but also incorporate it with temporal embeddings as conditional information for the diffusion model. It should be noted that ADA[31] involves only random transformations of images, and does not alter the unconditional nature of the model. It merely increases the diversity of the data during the training process. By utilizing this information, the model can better capture image features, enhancing its generalization ability and thereby generating higher-quality and more diverse synthetic datasets. Fig.2 illustrates our network architecture. First, time embeddings and ADA embeddings serve as conditional information for the diffusion model during the training process. The encoder performs downsampling on the noisy images, and after feature selection in the intermediate layers, the decoder performs upsampling to generate the final desired noise map, which is used for image denoising. For small-scale datasets, we use a three-layer architecture for the upsampling and downsampling modules. Algorithm 1 represents the preheating phase in our FSSL framework.

Algorithm 1 Global Distribution Data Generation Module(GDGM)

```
Input: Number of clients $K$, number of communication rounds $R$, number of local epochs $E$, local batch size $B$, local datasets $D_k$, learning rate $\eta$, number of diffusion steps $T$, variance schedule $\beta_1, ..., \beta_T$.
Output: Global diffusion model $\omega_r$
RunServer()
Initialization:Global model $\omega_0$
for each round $r = 1$ to $R$ do
    for each client $k = 1$ to $K$ do
        $\omega_r^k \leftarrow$ClientUpdate$(k, \omega_{r-1})$
    end for
    $\omega_r \leftarrow \frac{1}{|D|} \sum_{k=1}^{K} \omega_r^k \cdot |D_k|$
    if $r = R$ then
        $D_g \leftarrow$GenerateDataset$(\omega_r)$
        $\cup_{k=1}^{n+m} D_g^k \leftarrow D_g$
        for client $k = 1, 2, \dots, K$ do
            $D_k = D_k \cup D_g^k$
        end for
    end if
end for
ClientUpdate$(k, \omega_{r-1})$
$\omega_r^k \leftarrow \omega_{r-1}$
for $e = 1$ to $E$ do
    for $B \in D_k$ do
        $t \sim Uniform(\{1, \dots, T\})$
        $\epsilon_t \sim \mathcal{N}(0, I)$
        $\bar{\alpha}_t = \prod_{i=1}^{t} (1 - \beta_t)$
        $L_{dm} = \| \epsilon_t - \epsilon_{\omega_r^k}(\sqrt{\bar{\alpha}_t} i + \sqrt{1 - \bar{\alpha}_t} \epsilon_t, t) \|_2^2$
    end for
end for
Return $\omega_r^k$
```

3.3. Dynamic Class Balance Threshold Federated Semi-Supervised Learning

In FSSL, the limited labelled data and the existence of the non-IID problem lead to insufficient and rich learning of the model, which is very prone to catastrophic training bias. Therefore, we propose the dynamic class balancing threshold federal semi-supervised learning, which firstly can adaptively adjust the threshold of each class to ensure the richness of the samples, and at the same time can make the model screen more samples that can be assigned pseudo-labels in the early stage, which greatly alleviates the two major difficulties of FSSL. However, because of the uncertainty of pseudo-labels, we do not give up the rich knowledge contained in the low-confidence samples in the case of scarcity of real labelled data, and exploit the low-confidence samples using residual class negative learning while dynamically thresholding to differentiate between high-confidence samples and low-confidence samples. It can be said that our approach alleviates the main problem in FSSL to a great extent. We first introduce the optimization objectives for different types of clients, and then provide a detailed description of our approach.

During the training phase of federated learning, the training of fully labeled clients incorporates the unlabeled generated dataset Dg, making its update objective as $Ll = Ls + Lu1 + Lrnl$. For unlabeled clients, the training includes only unlabeled data, so their update objective is $Lu = Lu1 + Lrnl$.

For the supervised loss Ls, cross-entropy loss is used for training.

$$Ls = \frac{1}{B_l} \sum_{i=1}^{B_l} CE(f_\theta(y|\Omega(x_i^l)), y_i^l), \tag{10}$$

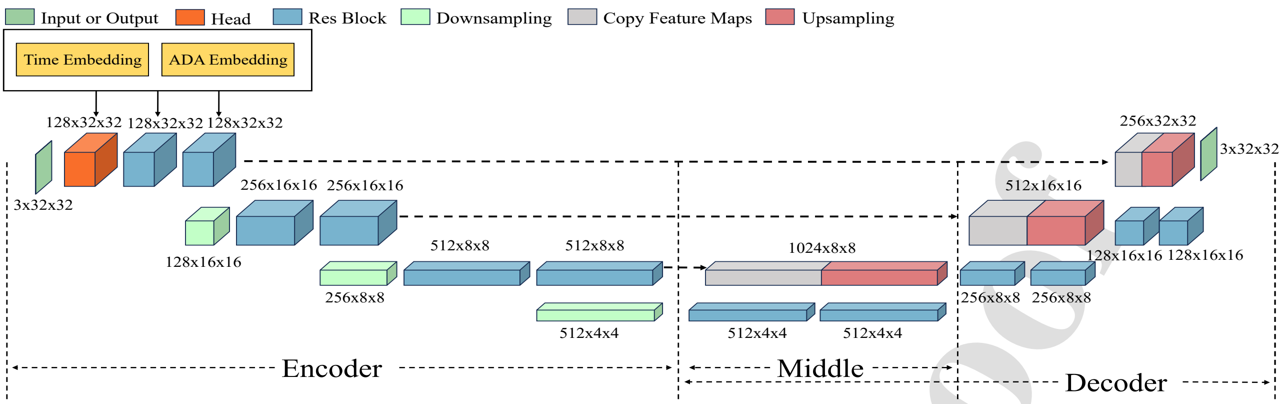


Fig. 2: Federated Learning Unet Architecture Diagram, which includes the encoder, decoder, and intermediate sections, and shows the feature variations under different operations, with temporal and ADA embeddings serving as guiding information for the model.

Here, B_l denotes the batch of labeled data, $CE(\cdot)$ represents the cross-entropy loss, f_θ is the local model, and $\Omega(\cdot)$ denotes the version of the data after performing DSA[32].

For the unsupervised loss L_{u1}, weakly augmented samples are used for training. The class with the highest predicted probability exceeding the threshold is taken as the pseudo-label, and cross-entropy loss is calculated based on this prediction.

$$L_{u1} = \frac{1}{B_u} \sum_{i=1}^{B_u} \mathbb{I}(\max(f_\theta(y|\Omega(x_i^u))) \geq \tau_t(c)) CE(\hat{y}_i^u, f_\theta(y|\Omega(x_i^u))), \tag{11}$$

Here, $\tau_t(c)$ denotes the threshold for class c at the current time step t, and $\hat{y}_i^u = \arg\max(f_\theta(y|\Omega(u_i)))$ represents the pseudo-label.

The residual class negative learning loss L_{rnl} will be elaborated upon in Section 3.3.2.

3.3.1. Dynamic Class-Balanced Threshold

In the field of SSL, methods such as pseudo-labeling have been developed to fully exploit the rich information contained in unlabeled data. The choice of threshold is critical to the quality of generated pseudo-labels, as overly high or low thresholds can adversely affect the model's performance. FreeMatch[18] adapts confidence thresholds dynamically based on the model's learning state but fails to address issues such as heterogeneous data partitioning in federated learning. CBAFed[5] designs a global class-balanced threshold by leveraging the data distribution information across all local clients. However, it is still constrained by a relatively high fixed threshold, which is not favorable during the early stages of training.

Inspired by FreeMatch[18] and CBAFed[5], we propose a novel confidence thresholding method, called Dynamic Class-Balanced Threshold (DCBT). Specifically, we integrate the model's current learning state with the difference between the global class distribution and standard deviation to dynamically adjust the confidence threshold for each class. This approach incrementally raises the threshold as the model learns and assigns different thresholds to classes at varying stages of learning. In the early stages of training, DCBT selects as many samples as possible to meet the diversity required for model training. In the later stages, it maintains a higher threshold, focusing on selecting high-quality samples.

First, we need to calculate the global class distribution for each communication round. For communication round r, the number of samples belonging to class c in labeled clients C_l and unlabeled clients C_u can be represented as follows:

$$\sigma_r^l(c) = \sum\nolimits_{i=1}^{N_l} \mathbb{I}(y_i^l = c), \tag{12}$$

$$\sigma_r^u(c) = \sum\nolimits_{i=1}^{N_u} \mathbb{I}(\max(q(u_i)) > \tau_t(c))\mathbb{I}(\hat{y}_i = c), \tag{13}$$

Here,$\mathbb{I}(\cdot)$ represents the indicator function. After completing local round r of training, clients upload their respective class distribution information $\sigma_r^l(c)$,$\sigma_r^u(c)$ and models to the server. The server then aggregates the total amount of data for class c in round r.

$$\sigma_r(c) = \sum\nolimits_{l=1}^{n} \sigma_r^l(c) + \sum\nolimits_{u=n+1}^{n+m} \sigma_r^u(c), \tag{14}$$

Subsequently, we normalize the data and compute the standard deviation of the global distribution.

$$\tilde{p}_r(c) = \frac{\sigma_r(c)}{\sum_{i=1}^{C} \sigma_r(i)}, \tag{15}$$

$$std(\tilde{p}_r) = \sqrt{\frac{1}{C-1} \sum\nolimits_{c=1}^{C} (\tilde{p}_r(c) - \bar{p}_r)^2}, \tag{16}$$

Thus, we propose a fundamental strategy to balance model confidence fluctuations by optimizing sample selection using the difference between the global class distribution and standard deviation. The global class distribution reflects the overall confidence level of the model for each class, which can represent the model's overall confidence in samples from that class to some extent. The standard deviation represents the stability of the model's predictions for samples in that class. A larger standard deviation indicates that the model's predictions for this class are less stable. By subtracting the standard deviation, we can avoid selecting samples with high confidence but large prediction fluctuations. Although the final threshold may be lower, the model will be more cautious in selecting samples. For classes with smaller standard deviations, this indicates that the model's confidence in these samples is more stable. The influence of the standard deviation on the threshold is reduced, allowing the model to confidently select high-confidence samples.

For the global threshold τ_t, we argue that a traditional high fixed threshold restricts the initial threshold, even when adjusted by the global class distribution and standard deviation difference, to a consistently high level. This limitation hinders improvements in the model's learning of minority classes. Therefore, we propose that τ_t should satisfy the following conditions: (1) It should increase progressively with the model's learning process. (2) It should reflect the model's learning status regarding the global unlabeled data. (3) It should quickly capture trends in data distribution changes under non-IID scenarios.

We first employ the exponential moving average (EMA) with momentum α to smooth the sample confidence over the current t step iteration, yielding the global threshold τ_t.

$$\tau_t = \alpha\tau_{t-1} + (1-\alpha)\frac{1}{B_u}\sum_{i=1}^{B_u} p_i, \tag{17}$$

Here, $\alpha \in (0,1)$ represents the momentum parameter of EMA, and p_i denotes the sample confidence. During each iteration, model updates are influenced by various factors such as data distribution and the accuracy of pseudo-labels. Although EMA can mitigate the negative effects of incorrect pseudo-labels, its inherent lag in the update mechanism limits the model's performance during sudden weight fluctuations. Compared to EMA, DEMA[33][34] assigns higher weight to the most recent data in the computation process, reducing lag caused by the influence of historical erroneous data. Therefore, we need to apply EMA to τ_t once more.

$$\tau'_t = \alpha\tau'_{t-1} + (1-\alpha)\tau_t, \tag{18}$$

Based on the DEMA formula $DEMA_t = 2 \cdot EMA_t - EMA(EMA_t)$, we derive the final calculation formula for the global threshold.

$$\tau_t = 2\tau_t - \tau'_t, \tag{19}$$

By combining the previously obtained global class distribution and its standard deviation, we can dynamically adjust the confidence threshold for each class based on the model's learning state using the following formula.

$$\tau_t(c) = \tau_t + \tilde{p}_r(c) - std(\tilde{p}_r). \tag{20}$$

where $\tilde{p}_r(c)$ denotes the global class distribution in the r-th round, and $std(\tilde{p}_r)$ denotes the standard deviation of the global data distribution in the r-th round.

3.3.2. Residual Class Negative Learning

In previous FSSL (Federated Semi-Supervised Learning) research, the training process typically involves selecting samples with the highest confidence exceeding a preset or dynamic threshold for training, while low-confidence samples are excluded. This approach often results in the abandonment of many samples during the early stages of training, failing to utilize the potential information in all samples. Additionally, in later stages, the model may become overly reliant on high-confidence samples, focusing only on easily learnable data, ultimately leading to reduced generalization ability. Under complex data distributions, such selective training may fail to cover the diversity of the sample space.

To address this issue and fully leverage the knowledge in low-confidence samples, we propose the Residual Class Negative Learning (RNL) method, inspired by UDA[35], NLNL[36], and FullMatch[37]. In traditional semi-supervised learning, models are typically trained to focus only on learning the correct labels. In contrast, negative learning directs the model to recognize incorrect categories. By training the model to identify categories to which a sample does not belong, negative learning helps reduce misclassification and enhances the model's discriminative ability. Our hypothesis is that when the $top-n$ accuracy of the consistency of the different augmented versions of a sample is high enough, there is a high probability that the sample belongs to one of the n classes, and we can optimize the model's ability by suppressing the other $C - n$ classes. As shown in the Fig. 3, when the true class of a sample is uncertain and the model makes an ambiguous decision on high-confidence categories (blue predictions), negative pseudo labels can be assigned to low-confidence categories (red predictions) to refine the decision boundaries. Given that we have obtained an optimal value through computational accuracy, applying negative pseudo labels to low-confidence categories is a reliable and effective approach. Negative pseudo labels play a crucial role in our method, particularly in enhancing model robustness, as they explicitly indicate which categories a sample does not belong to. Due to this unique mechanism, negative pseudo labels enable all unlabeled data to participate in the learning process, thereby improving data utilization.

Unlike FullMatch[37], which focuses solely on centralized SSL and ignores heterogeneous data partitioning, our approach, under the FSSL paradigm with non-IID data partitioning, optimizes computational efficiency by dividing the training of unsupervised clients in federated semi-supervised learning into two parts: learning from high-confidence samples selected via dynamic class-balanced thresholds and learning from low-confidence samples. For high-confidence samples, we assume that the model has already developed strong discriminative ability. Thus, after generating unified pseudo-labels using the global model, we proceed with standard training. For low-confidence samples, we collect two different augmented versions and first compute their $top-n$ accuracy based on consistency between the

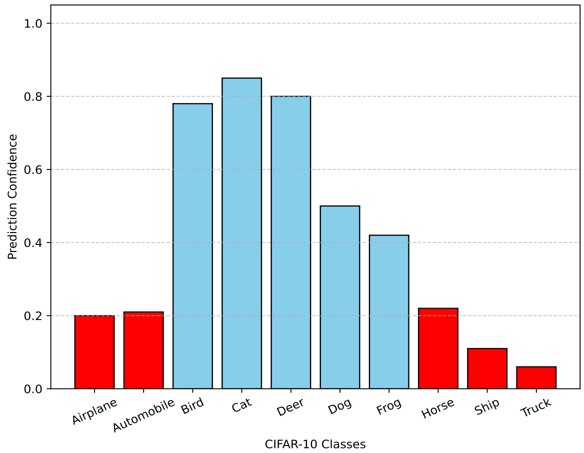


Fig. 3: An example of negative pseudo-label assignment on the CIFAR-10 dataset

augmented versions. The remaining $C - n$ low-confidence classes are then assigned negative pseudo-labels for suppression. This split training approach helps the model better handle decision boundaries in ambiguous data, avoiding the complete exclusion of low-confidence samples and improving overall model performance.

However, two major challenges arise: (1) how to evaluate $top - n$ accuracy during training and (2) how to determine the value of n. Regarding the first challenge, evaluating top-n accuracy typically requires an additional dataset, and using a validation set increases computational overhead. During the early data processing stage, we apply strong and weak augmentations to all samples. Due to the dynamic threshold, augmented versions of low-confidence samples are often discarded. We propose using these augmented versions as an additional dataset and optimizing the consistency between two different augmented versions to reflect the model's performance. For the second challenge, we believe that n should dynamically change as the consistency between strong and weak augmented versions improves. In the early stages of training, the consistency difference between augmented versions is relatively large, resulting in a larger n. In later stages, as consistency improves, n gradually decreases. If n is fixed (e.g., $top - 5$ accuracy), this would mean that at all training stages, we only evaluate the top-5 accuracy and suppress the confidence of the remaining 5 classes. Such a rigid approach may degrade model performance, especially during the early stages of training.

Addressing these two challenges, our specific approach first derives a pseudo-label $\tilde{y}$ based on the prediction results of the weakly augmented version,

$$\tilde{y} = \arg\max(\max(f_{\theta}(y|\Omega(x_i^u))) < \tau_t(c)), \tag{21}$$

Next, we sort the outputs of the strongly augmented version $f_{\theta}(y|\alpha(x_i^u))$ in descending order and record the category indices, obtaining a vector $s(i)_c$ that represents the categories ranked by confidence. For each $\tilde{y}$, we check its position in this ranking. For the batch of samples, we calculate a Boolean tensor satisfying $\mathbb{I}(\sum_{c=1}^{C} s(i = \tilde{y})_c)$ for each class and compute the accuracy. The smallest class index that ensures the accuracy is at least 99.9% is the desired n,

$$n = \arg\min_{c \in [1,C]}(acc(\tilde{y}, s(i)_c) \geq 99.9\%), \tag{22}$$

The calculation method for accuracy rate is as follows:

$$acc(\tilde{y}, s(i)_c) = \frac{1}{B_u}(\mathbb{I}(\sum_{c=1}^{C} s(i = \tilde{y})_c)) \cdot 100\%), \tag{23}$$

We have determined the n value that satisfies the condition for low-confidence samples. Next, we assign negative pseudo-labels to the classes in the weakly augmented samples' outputs, excluding the top n classes with the highest confidence. Similarly, we sort the weakly augmented samples' outputs by confidence to obtain the vector $w(i)_c$. After assigning the negative pseudo-labels, the weakly augmented output vector $w(i)_c$ becomes:

$$w(i)_c = \mathbb{I}(w(i)_c > n), \tag{24}$$

Similar to NLNL[36], our residual class negative learning loss can be defined as:

$$L_{rnl} = -\frac{1}{B_u} \sum_{i=1}^{B_u} \sum_{c=1}^{C} \mathbb{I}(w(i)_c > n) \log(1 - s(i)_c). \tag{25}$$

Defining such a loss function can be understood as follows: during the training process, the closer the model's prediction is to the negative pseudo-label $\mathbb{I}(w(i)_c > n)$, the closer the term approaches 0. This increases the loss, forcing the model to develop the ability to distinguish incorrect labels, thereby improving its generalization. Additionally, in the early stages of training, cases where $n = C$ may occur. This means that the class with the highest confidence in the weakly augmented sample's output has the lowest confidence in the strongly augmented version, resulting in all classes being part of the $top - n$. Consequently, no loss would be generated.

Our residual class negative learning (RNL) abandons the traditional resource and computation-intensive approach of using a validation set to evaluate $top - n$ accuracy. Instead, it cleverly optimizes the model by leveraging the consistency between different augmented versions of the same samples. Furthermore, it is the first method to incorporate the utilization of low-confidence data into the FSSL domain, significantly enhancing the model's generalization and performance.

3.3.3. Aggregation Method

Due to the non-IID setting, the simple federated averaging aggregation method can lead to the accumulation of incorrect knowledge in models on unlabeled clients, exacerbating model drift between clients. Therefore, we increase the weight of labeled clients during model aggregation to correctly guide the learning direction of the model. Additionally, since the generated dataset helps mitigate data heterogeneity among clients, we propose merging the high-confidence samples (above the threshold) from both the original and generated datasets on each client in every round for aggregation. Low-confidence samples are retained solely for local training on each client to optimize local models.

$$\delta_t^k = \begin{cases} \dfrac{|D_k + D_{g,t}^k|}{|D_t^{train}|} & if\ k \in \{1, \dots, n\} \\ \dfrac{|D_{k,t}^{train} + D_{g,t}^k|}{|D_t^{train}|} & if\ k \in \{n+1, \dots, n+m\} \end{cases}, \tag{26}$$

Here, $|D_{g,t}^k|$ represents the number of samples selected from the generated dataset by client C_k in the t communication round for training, along with $|D_t^{train}| = \sum_{l=1}^n |D_l + D_{g,t}^l| + \sum_{u=n+1}^{n+m} |D_{u,t}^{train} + D_{g,t}^u|$.In this way, we obtain the coefficient related to client data size for model aggregation, and the global model can be calculated as follows:

$$\theta_{t+1}^{glob} = \sum_{k=1}^{n} \varphi \delta_t^k \theta_t^k + \sum_{k=n+1}^{n+m} \bar{\delta}_t^k \theta_t^k. \tag{27}$$

θ_{t+1}^{glob} represents the global model at the $t+1$ communication round, and φ is the coefficient that adjusts the weight of labeled clients. Our dynamic class-balanced threshold federated semi-supervised learning process is detailed in Algorithm 2.

4. Experiments

In this section, we first introduce the datasets and related settings used in the experiments, followed by an explanation of the experimental setup, comparative experiments, and ablation studies for DDRFed.

4.1. Experimental setup

4.1.1. Data set and partition settings

To evaluate our proposed DDRFed framework, we conducted experiments on the CIFAR-10 and Fashion-MNIST datasets. The CIFAR-10 dataset consists of 50,000 training samples and 10,000 testing samples, divided into ten categories of natural images, with a sample size of 32×32. Fashion-MNIST contains 60,000 training samples and 10,000 testing samples, consisting of ten categories of grayscale fashion product images, with a sample size of 28×28. We maintain the original dataset structure to set up the training and testing sets, facilitating the comparison of different methods. The same preprocessing steps are applied to both datasets. For weak augmentation, we randomly flip the images horizontally with a 50% probability, apply reflection padding of 4 pixels, and then crop them back to the original size. For strong augmentation, we apply three random transformations (such as rotation and color changes) with an intensity level of 5, based on the weak augmentation.

For dataset partitioning, we adopted the same strategy as RSCFed[4] and CBAFed[5]. The Dirichlet distribution $Dir(\gamma)$ was used to perform non-IID partitioning across clients. Smaller values of γ indicate more imbalanced data distributions across clients, resulting in varying class coverage and sample counts for each client. When $\gamma = 0.8$, we set the ratio of labeled to unlabeled clients to 1:9. For $\gamma = 0.1$, to ensure the effectiveness of the classification task, we adopted a 5:5 ratio.

Algorithm 2 DDRFed

```
Input: Number of clients $K$,number of communication rounds $T$,global model $\theta_t^{glob}$,learning rate $\eta$,Generated dataset $D_g^k$,weight coefficient $\varphi$,label batch $B_L$,unlabel batch $B_u$
Output: Global model $\theta_t^{glob}$
RunServer()
Initialization:Global model $\theta_0^{glob}$
for each round $t = 1, 2, \dots, T$ do
    for each client $k = 1, 2, \dots, K$ do
        send global model $\theta_t^{glob}$
        $D_k, \sigma_{t+1}^k, \theta_{t+1}^k \leftarrow ClientUpdate(k, \sigma_t, \theta_t^{glob})$
        Calculate $\delta_t^k$
    end for
    $\sigma_{t+1} = \sum_{k=1}^{n+m} \sigma_{t+1}^k(1), ..., \sum_{k=1}^{n+m} \sigma_{t+1}^k(C)$
    $\theta_{t+1}^{glob} = \sum_{k=1}^{n} \varphi \delta_t^k \theta_t^k + \sum_{k=n+1}^{n+m} \bar{\delta}_t^k \theta_t^k$
end for
$ClientUpdate(k, \sigma_t, \theta_t^{glob})$
if $k < n$ then
    for each batch $B_L$ from $D_L$ and $B_u$ from $D_g^k$ or $D_u$ do
        $\tau_t = \alpha \tau_{t-1} + (1-\alpha) \frac{1}{B_u} \sum_{i=1}^{B_u} p_i$
        $\tau_t = 2\tau_t - \tau'_t,\ where\ \tau'_t = \alpha \tau'_{t-1} + (1-\alpha)\tau_t$
        $\tau_t(c) = \tau_t + \tilde{p}_r(c) - std(\tilde{p}_r)$
        $\sigma_t^k(c) = \sigma_t^k(c) + \sum_{i=1}^{B_u} \mathbb{I}(\max(q(u_i)) > \tau_t(c))\mathbb{I}(\hat{y}_i = c)$
        $\tilde{y} = \arg\max(\max(f_\theta(y|\Omega(x_i^u))) < \tau_t(c))$
        $n = \arg\min_{c \in [1,C]}(acc(\tilde{y}, s(i)_c) \geq 99.9\%)$
        $L_{u1} = \frac{1}{B_u} \sum_{i=1}^{B_u} \mathbb{I}(\max(f_\theta(y|\Omega(x_i^u))) \geq \tau_t(c)) CE(\hat{y}_i^u, f_\theta(y|\alpha(x_i^u)))$
        $L_{rnl} = -\frac{1}{B_u} \sum_{i=1}^{B_u} \sum_{c=1}^{C} \mathbb{I}(w(i)_c > n) \log(1 - s(i)_c)$
        $L = L_s + L_{u1} + L_{rnl}$
        $\theta_t^k \leftarrow \theta_t^k - \eta \nabla L$
    end for
else
    $L_u = L_{u1} + L_{rnl}$
end if
Return $\sigma_t^k, sum(\sigma_t^k), \theta_t^k$
```

4.1.2. Experimental details

To ensure fairness in comparing various FSSL methods, we used the same network structure and experimental setup across all datasets, including client ratios, data volumes, and optimizers. We utilized ResNet18 from the PyTorch framework as the backbone network and employed an SGD optimizer with a momentum of 0.9. The learning rates were set to 0.03 for labeled clients and 0.02 for unlabeled clients, with a batch size of 64. Each client had identical data partitioning strategies and data volumes, and experiments were conducted on NVIDIA Tesla V100 GPU. For the federated diffusion training stage, specifically the warm-up phase, we

Table 1
Experimental Results for CIFAR-10 and Fashion-MNIST Datasets

Labeling Strategy	Method	CIFAR-10				Fashion-MNIST			
		Dir($\gamma = 0.1$)		Dir($\gamma = 0.8$)		Dir($\gamma = 0.1$)		Dir($\gamma = 0.8$)	
		ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Fully supervised	FedAvg[1] (upper-bound)	62.60	93.14	87.12	98.99	84.92	98.82	91.13	99.44
	FedAvg[1] (lower-bound)	48.41	91.33	57.92	91.68	59.56	96.92	81.96	98.24
Semi supervised	FedIRM[6]	53.07	92.02	59.37	91.92	59.35	96.90	78.16	97.17
	Fed-Consist[23]	50.15	93.38	56.52	90.15	57.38	96.33	81.93	97.99
	RSCFed[4]	49.04	91.26	54.64	89.59	57.76	96.97	79.88	98.38
	PDCFed[24]	51.50	91.61	55.01	89.85	61.21	95.15	76.37	97.18
	CABFed[5]	53.13	93.61	62.16	90.79	64.52	95.89	83.37	97.59
	DDRFed (ours)	**59.51**	**94.12**	**69.09**	**94.93**	**72.45**	**97.06**	**85.54**	**98.90**

trained the unconditional diffusion model for 40 rounds. In the formal federated learning phase, we first conducted 200 rounds of pretraining on the labeled clients, followed by 500 rounds of federated communication.

4.2. Comparison experiment

4.2.1. Methods of comparison

Under non-IID partitioning conditions, we compared DDRFed with state-of-the-art FSSL methods. These include: (1) FedIRM[6], which links the learning of labeled and unlabeled clients through a client relationship matching scheme. (2) Fed-Consist[23], which integrates consistency regularization and pseudo-labeling into FSSL. (3) RSCFed[4], which enhances model performance by subsampling clients and using distance-weighted aggregation. (4) CBAFed[5], which introduces class-balanced adaptive thresholds and residual weight connections to mitigate non-IID issues. (5) PDCFed[24], which proposes a two-stage sampling method by predicting distribution shifts of different data augmentations. Additionally, to better define the performance upper and lower bounds of the models, we followed the FedAvg[1] approach, treating the case with 10 labeled clients as the upper bound and the case with 1 labeled client as the lower bound. We used accuracy and the area under the ROC curve (AUC) as evaluation metrics.

4.2.2. Comparison of experimental results

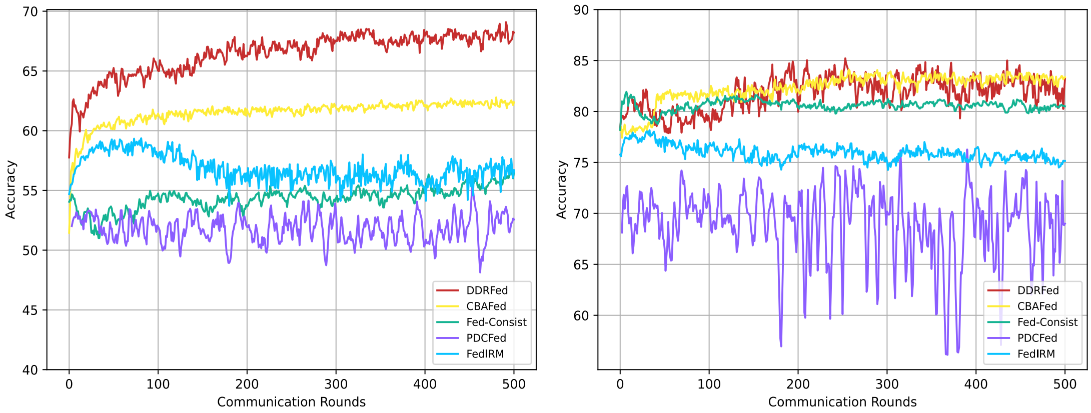
Table 1 presents the comparative experimental results of various methods on the CIFAR-10 and Fashion-MNIST datasets, implemented according to the experimental details. Here, γ represents the parameter controlling the degree of data heterogeneity; the smaller γ, the more imbalanced the data distribution. Regardless of the data partitioning scenario, our DDRFed method consistently outperforms all other federated semi-supervised methods.

It can be observed that when the data distribution among clients is relatively balanced ($\gamma = 0.8$), our method improves accuracy on the CIFAR-10 dataset by 6.93% and AUC by 4.14% compared to state-of-the-art methods. On the Fashion-MNIST dataset, accuracy increases by 2.17% and AUC by 1.31%.When the client data distribution is extremely imbalanced ($\gamma = 0.1$), our method also significantly outperforms all other methods. On the CIFAR-10 dataset, accuracy improves by 6.38% and AUC by 0.51%.On the Fashion-MNIST dataset, accuracy and AUC are improved by 7.93% and 1.17%, respectively. Furthermore, Fig.4 illustrates the accuracy curve of our DDRFed method compared to other methods on CIFAR-10 and Fashion-MNIST. It can be seen that DDRFed converges faster and consistently maintains a high level of accuracy compared to other methods. The reason lies in the fact that, although CBAFed[5] has achieved relatively good performance due to its thresholding method, it is still constrained by a fixed base threshold. This limitation causes the class thresholds in CBAFed[5] to fluctuate only within a relatively high range, making it less favorable for early-stage training. PDCFed[24] and RSCFed[4], due to their client sampling training approach, cannot ensure that labeled clients participate in every training round, making it impossible to effectively guide model training. Fed-Consist[23] and FedIRM[6] do not thoroughly address the non-IID problem in FSSL, leading to suboptimal performance. In contrast, our DDRFed not only addresses the non-IID problem using a diffusion model and the flexible, non-fixed-threshold DCBT but also enhances the model's learning capability through RNL, resulting in the best overall performance.

4.2.3. Number of unlabeled clients

In order to observe the model performance of our method with different proportions of labeled and unlabeled clients, we conducted experiments on the CIFAR-10 dataset with a total number of clients of 10, where there are 2 labeled and 8 unlabeled clients, and a non-IID partition setting of $\gamma = 0.8$. The results of our comparison are shown in Table 2. It can be seen that our method significantly outperforms other methods, improving 4.54% and 1.65% in accuracy and AUC, respectively, compared to the state-of-the-art method.

By comparing Table 1 and Table 2, we can conclude that the model's performance enhancement is most effective



(a) (b)

Fig. 4: Accuracy curves of our FSSL method versus other methods on CIFAR-10 and Fashion-MNIST datasets.

Table 2
Experimental results for the CIFAR-10 dataset when the number of unlabeled clients is 8.

Method	Client Num		ACC	AUC
	Labeled	Unlabeled		
FedAvg[1] (upper-bound)	10	0	87.12	98.99
FedAvg[1] (lower-bound)	2	0	67.83	94.85
FedIRM[6]	2	8	67.14	93.61
Fed-Consist[23]	2	8	71.25	95.41
RSCFed[4]	2	8	66.46	94.64
PDCFed[24]	2	8	66.93	93.92
CABFed[5]	2	8	71.83	95.38
DDRFed (ours)	2	8	**76.37**	**97.03**

when the total number of clients is constant and the percentage of unlabeled clients is 90% of the total. This proves the significant effectiveness of our approach for the utilization of unlabeled data.

4.2.4. Higher number of clients

In order to evaluate the performance of our method with a larger number of clients, we conducted ablation studies of our methods, CBAFed[5] and RSCFed[4], respectively, on the CIFAR-10 dataset in the case of . First we set the total number of clients to 5, 10, 20, 30 and 50, and make the percentage of unlabeled clients set to 80%. Table 3 represents our experiments on RSCFed[4], CBAFed[5] and DDRFed with different total number of clients. From Table 3, we can see that our method improves its accuracy and AUC by 3.62% and 0.81%, respectively, compared with the state-of-the-art FSSL method in the case of a small total number of clients, such as 5. In the case of a total number of clients, 50, our method also makes full use of the knowledge of the unlabeled data of each client, which improves the accuracy and AUC by 6.38% and 2.87%, respectively. 6.38% and 2.87%. And by comparing the experimental performance of each case and each FSSL method side-by-side, our method always obtains the most stable improvement.

The quantitative analysis above demonstrates that the performance of our DDRFed method outperforms all other methods across varying levels of data heterogeneity. Particularly when the data heterogeneity is severe($\gamma = 0.1$),our method fully leverages the generated dataset assigned by GDGM for each client, and uses DCBT to lower the threshold for minority classes, addressing the class imbalance issue, which significantly alleviates the class imbalance problem. Finally, RNL is used to leverage low-confidence samples to optimize each local client model, ensuring that the aggregated model does not experience significant model drift.

4.3. Ablation experiment

In order to explore the effects of all three of Residual Class Negative Learning (RNL), Dynamic Class Balance Thresholding (DCBT) and Global Distribution Data Generation (GDGM) as well as various hyper-parameters on DDRFed, we conducted ablation experiments and analyzed each module.

4.3.1. Contribution of GDGM, DCBT and RNL

In order to more intuitively observe the impact of different FSSL methods on model performance and to validate the advantages of the DDRFed method, we choose Fed-Consist[23] as our baseline method. Table 4 presents the quantitative experimental results on two datasets, where we can observe that DCBT, RNL, and DG all contribute to the model's performance. By comparing the quantitative results of "Basic" and "Basic+DCBT," we observe that DCBT improves the accuracy and AUC on the CIFAR-10 dataset by 8.86% and 2.32%, respectively, and on the Fashion-MNIST dataset by 2.52% and 0.42%, respectively. Next, our RNL improves the accuracy and AUC on the CIFAR-10 dataset by 1.45% and 0.13%, respectively. For the Fashion-MNIST dataset, the use of DCBT has already significantly improved the model's performance, enabling

Table 3
Experiments with Different Number of Clients.

Total numbers	Client Num		RSCFed[4]		CBAFed[5]		DDRFed(ours)	
	Labeled	Unlabeled	ACC	AUC	ACC	AUC	ACC	AUC
5	1	4	61.21	91.87	64.52	92.36	**68.14**	**93.17**
10	1	9	54.64	89.59	62.16	90.79	**69.09**	**94.93**
20	2	18	60.69	92.69	66.78	93.34	**71.70**	**96.01**
30	3	27	62.17	92.33	68.89	93.48	**74.52**	**96.32**
50	5	45	61.55	92.49	65.66	92.93	**72.04**	**95.80**

Table 4
Experimental Results for CIFAR-10 and Fashion-MNIST Datasets

Method	DCBT	RNL	GDGM	Metrics	
				ACC	AUC
				Dataset 1: CIFAR-10	
Basic				56.52	90.15
Basic+DCBT	✓			65.38	92.47
Basic+DCBT+RNL	✓	✓		66.83	92.60
DDRFed(ours)	✓	✓	✓	**69.09**	**94.93**
				Dataset 2: Fashion-MNIST	
Basic				81.93	97.99
Basic+DCBT	✓			84.45	98.41
Basic+DCBT+RNL	✓	✓		84.52	98.48
DDRFed(ours)	✓	✓	✓	**85.54**	**98.90**

it to make better decisions for each class of samples. As a result, in each round of federated communication, the number of unselected low-confidence samples is minimal, thus reducing the contribution of RNL to the model. Finally, our GDGM module increases the accuracy and AUC on the CIFAR-10 dataset by 2.26% and 2.33%, respectively, and on the Fashion-MNIST dataset by 1.02% in accuracy and 0.42% in AUC. The above experimental data show that assigning a diffusion model-generated dataset to each client and then using dynamic class balancing thresholds to divide the training into high-confidence and low-confidence samples makes a significant contribution to FSSL.

4.3.2. *A study of a globally distributed data generation module*

In order to better understand the contribution of the globally distributed data generation module to DDRFed, we first compare the original and diffusion images of the two datasets, as shown in (a) and (b) in Fig.5, where the blue line above indicates the original image and the diffusion image below, and it can be seen that the diffusion model combined with the federated learning training approach generates images that not only satisfy global data distributions, the Moreover, their quality and contained knowledge are already comparable to the original images, which can be fully supplemented to the client to alleviate the non-IID problem. Then, about the size of the generated dataset we have also explored in this paper, if the generated dataset is too small, it may not be able to effectively contribute to the model to mitigate the non-IID problem, and if the generated dataset is too large, it will result in unnecessary wastage of computational resources, Fig. 6 represents our exploration of the impact of generating dataset size on model performance on the CIFAR-10 dataset. The model's performance is most superior when the size of the generated dataset is one times of the training set, so from the perspective of performance and saving of computational resources, all our experiments on DDRFed have the generation coefficients ψ all of 1.

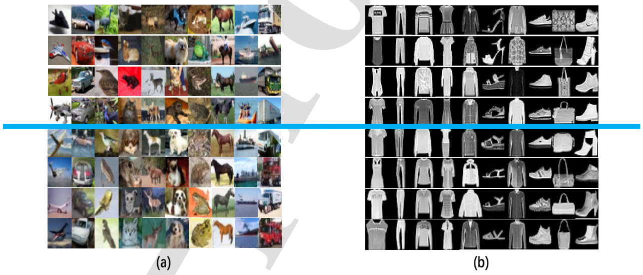


Fig. 5: (a) and (b) are the CIFAR-10 and Fashion-MNIST dataset images, respectively, where the original image is above the blue line and the diffusion model generated image is below the blue line.

4.3.3. *A study of Dynamic Class-Balanced Threshold and Residual Class Negative Learning*

DDRFed filters out high and low confidence samples by utilizing Dynamic Class-Balanced Threshold (DCBT) and exploits low confidence samples through Residual Class Negative Learning (RNL) to improve model performance. To explore our thresholding approach compared to other state-of-the-art methods, we migrated FreeMatch[18] and FixMatch[16] to the FSSL scenario and compared them with our approach, the comparison results are shown in Table 5 . Our method significantly outperforms the state-of-the-art FreeMatch[21] method with 6.47% accuracy and 1.05% AUC improvement on the CIFAR-10 data and 3.74%

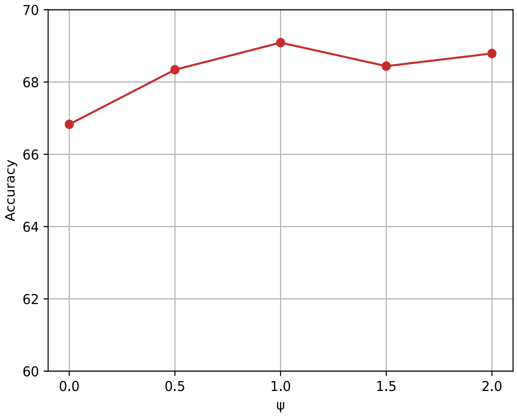


Fig. 6: The effect of the time generation factor on the accuracy in the CIFAR-10 dataset. The size of the coefficients represents the total amount of client data generated for the corresponding multiplier.

accuracy and 0.39% AUC improvement on the Fashion-MNIST dataset. Fig. 7(a) and (b) show the trends of threshold changes at the server for the CIFAR-10 dataset with non-IID settings of $\gamma = 0.8$ and $\gamma = 0.1$, respectively. It is evident from the figures that, in the early training stages, the thresholds for all categories continuously increase. As the model's learning ability improves, the threshold adjustments become dynamically optimized based on the model's learning status for each category. Under highly imbalanced data distributions (Fig. 8(b)), the thresholds are maintained at lower levels to accommodate the challenges posed by data imbalance. FixMatch[16] employs a fixed and relatively high threshold, which may hinder the model's effective learning of samples in the early training stages. In contrast, FreeMatch[18] does not account for data partitioning and heterogeneity, resulting in an excessively slow threshold increase. This leads to frequent label prediction errors in the early stages of FSSL training, adversely affecting subsequent learning performance.

We also explored the effect on performance of the augmented version of the data used to predict the temporary pseudo-label $\tilde{y}$ in RNL. As shown in Table 5,When we used strong augmented to predict the temporary pseudo-label and examined the position of $\tilde{y}$ in the weakly augmented data for residual class negative learning, the model performance degraded because the strongly augmented version has a larger perturbation compared to the original image, and the model may make more incorrect predictions in the pre-training period. The model performance becomes suboptimal when we fix the n value and only compute the $top{-}5$ accuracy of the low confidence data in each round of training for each client, because of the inaccuracy of the temporary pseudo-labeling, the low confidence data may just happen to be the other five residual classes, which will result in our residual class negative learning not contributing to the model performance improvement.

Table 5
Comparison experiments of different semi-supervised methods,ours(n=5) represents that we fix the value of top-n, and ours(change) represents that we use a strongly enhanced version to predict temporary pseudo-labels.

Methods	CIFAR-10		Fashion-MNIST	
	ACC	AUC	ACC	AUC
FixMatch[16]	59.43	90.75	79.23	97.91
FreeMatch[18]	60.36	91.55	80.78	98.09
ours (n=5)	62.67	92.65	84.08	98.21
ours (change)	65.92	92.04	84.15	98.27
ours (DCBT+RNL)	**66.83**	**92.60**	**84.52**	**98.48**

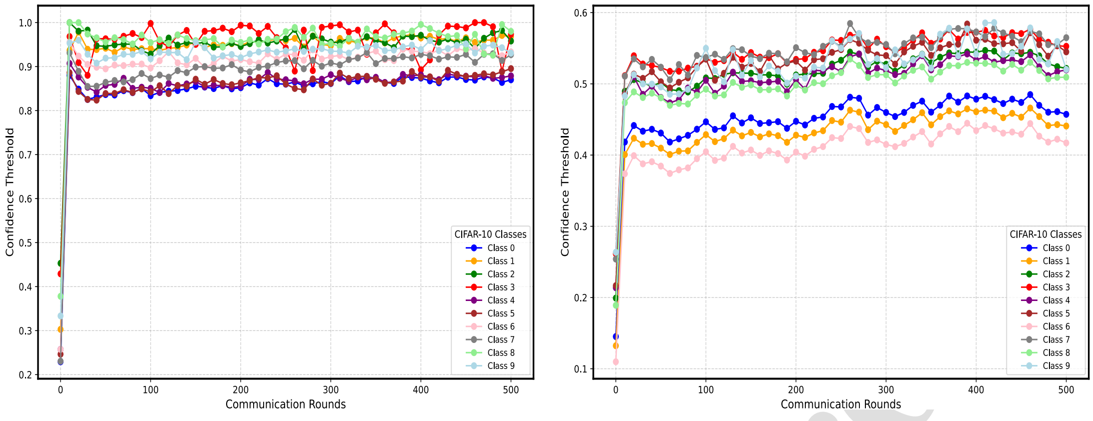
4.3.4. Communication cost

We compared the communication costs of Fed-Consist[23], RSCFed[4], and CBAFed[5] with DDRFed. To ensure an equal number of participating clients in each federated training round, we employed a two-round subsampling strategy in RSCFed[4], selecting five clients per round. In DDRFed, each round requires an exchange of data category distribution information between clients and the server. However, due to its minimal size, this exchange is negligible and does not significantly impact overall communication overhead.In the GDGM phase, we use the FedAvg[1] to train the diffusion model. In a single communication round with 10 clients, the communication overhead is approximately 2.65GB. As the number of clients or training rounds increases, computational resource consumption becomes substantial. To reduce communication overhead, we conducted only 40 rounds of federated communication in the globally distributed data generation phase and performed an additional 10 rounds of local training on clients to stabilize the model. This approach reduces the communication overhead of the diffusion model to only one-fourth of that in the federated semi-supervised learning phase, significantly lowering computational and communication costs while ensuring performance, making the method more feasible in practical applications.

Table 6 presents the communication costs at different stages, where "Diffusion" refers to the communication overhead of the GDGM, and "Classification" represents the communication cost during the federated semi-supervised learning phase. As shown in Table 6, DDRFed incurs higher communication costs than the other compared methods under the same number of clients but achieves significant performance advantages. Furthermore, when the number of clients is halved, DDRFed still achieves substantial performance improvements with lower communication costs, further demonstrating its superior balance between communication efficiency and model performance.

4.3.5. Hyperparameter Sensitivity Analysis

We examine the effects of several critical hyperparameters on DDRFed's performance.Fig. 8 represents the sensitivity analysis of our method for hyperparameters.



(a)

(b)

Fig. 7: Threshold variation trends for different non-IID settings on the CIFAR-10 dataset.

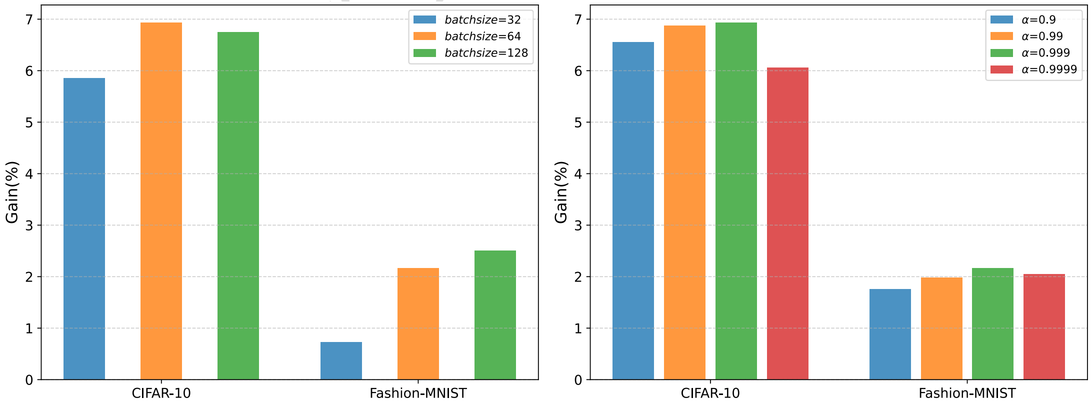
Table 6
Comparison of communication overhead for different methods on the CIFAR-10 dataset

Method	Client Num.	Com. Cost (GB)		Metrics	
		Diffusion	Classification	ACC	AUC
Fed-Consist [23]	10	–	419.92	56.52	90.15
RSCFed [4]	10	–	419.92	54.36	89.37
CBAFed [5]	10	–	419.92	62.16	90.79
DDRFed	5	53.13	209.96	68.14	93.17
DDRFed (ours)	10	106.25	419.92	**69.09**	**94.93**

Batch Size:The batch size can greatly influence model performance. We experimented with batch sizes of 32, 64, and 128. Fig. 8(a) illustrates that smaller batch sizes lead to inferior performance, while medium and large batch sizes improve performance across both datasets. These results indicate that batch size plays a key role in determining model performance.

DEMA Momentum α:The magnitude of the DEMA momentum may affect DCBT's control over class thresholds and the model's sensitivity to new parameters. Therefore, we conducted experiments with DEMA momentum values of 0.9, 0.99, 0.999, and 0.9999. As shown in Fig. 8(b), the model exhibits little sensitivity to the DEMA momentum values across both datasets.

In summary, DDRFed effectively mitigates the non-IID problem for small datasets. However, due to the complexity of large-scale scenarios and the inherent limitations of unconditional diffusion models, the GDGM module requires



(a) batch size

(b) α

Fig. 8: An ablation study on the proposed method concerning batch size, DEMA momentum α. The accuracy gains achieved by DDRFed over CBAFed[5] are recorded.

a large amount of computational resources and time, and the training and sample generation time often exceeds two weeks. To address the challenges faced by diffusion models in joint learning, future research will focus on improving the training efficiency of diffusion models and reducing the computational overhead.

5. Conclusion

In this paper, we propose a federated semi-supervised learning framework that leverages diffusion models, dynamic class-balanced thresholds, and residual class negative learning. Experimental results demonstrate that powerful diffusion models and thresholding methods that adjust based on model learning status can effectively address the non-IID problem in FSSL. Additionally, improving the utilization of unlabeled data strengthens model learning, mitigating the adverse effects of limited labeled data in FSSL. In future work, we will focus on ways to better integrate diffusion modeling and federated learning. In addition we will conduct experiments on larger and realistic datasets to demonstrate the generalization and effectiveness of DDRFed.

6. Acknowledgement

This work received partial support from the National Natural Science Foundation of China (NSFC) under Grants 61861004 and 61962004, along with funding from the Finnish Cultural Foundation for North Ostrobothnia Regional Fund (Grant 60231712), the Instrumentarium Foundation (Grant 240016) and the Innovation Project of Guangxi Graduate Education(Grant YCSW2024112). (Corresponding author: Feng Yang.)

CRediT authorship contribution statement

Zeyuan Wang: Writing-review & editing, Writing-original draft, Methodology, Software, Validation, Conceptualization. **Yang Liu:** Writing-review & editing, Supervision, Investigation. **Guirong Liang:** Writing-review & editing, Supervision, Investigation, Data curation. **Cheng Zhong:** Writing-review & editing, Supervision, Conceptualization. **Feng Yang:** Writing-review & editing, Supervision, Data curatioin, Conceptualization.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial intelligence and statistics, PMLR, 2017, pp. 1273–1282.

[2] W. Jeong, J. Yoon, E. Yang, S. J. Hwang, Federated semi-supervised learning with inter-client consistency & disjoint learning, arXiv preprint arXiv:2006.12097 (2020).

[3] H. Lin, J. Lou, L. Xiong, C. Shahabi, Semifed: Semi-supervised federated learning with consistency and pseudo-labeling, arXiv preprint arXiv:2108.09412 (2021).

[4] X. Liang, Y. Lin, H. Fu, L. Zhu, X. Li, Rscfed: Random sampling consensus federated semi-supervised learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 10154–10163.

[5] M. Li, Q. Li, Y. Wang, Class balanced adaptive pseudo labeling for federated semi-supervised learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 16292–16301.

[6] Q. Liu, H. Yang, Q. Dou, P.-A. Heng, Federated semi-supervised medical image classification via inter-client relation matching, in: Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III 24, Springer, 2021, pp. 325–335.

[7] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Advances in neural information processing systems 33 (2020) 6840–6851.

[8] J. Song, C. Meng, S. Ermon, Denoising diffusion implicit models, in: International Conference on Learning Representations.

[9] H. Zhu, J. Xu, S. Liu, Y. Jin, Federated learning on non-iid data: A survey, Neurocomputing 465 (2021) 371–390.

[10] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Srivastava, A survey on security and privacy of federated learning, Future Generation Computer Systems 115 (2021) 619–640.

[11] J. Konečnỳ, Federated learning: Strategies for improving communication efficiency, arXiv preprint arXiv:1610.05492 (2016).

[12] Q. Li, B. He, D. Song, Model-contrastive federated learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 10713–10722.

[13] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, A. T. Suresh, Scaffold: Stochastic controlled averaging for federated learning, in: International conference on machine learning, PMLR, 2020, pp. 5132–5143.

[14] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, Proceedings of Machine learning and systems 2 (2020) 429–450.

[15] H.-Y. Chen, W.-L. Chao, Fedbe: Making bayesian model ensemble applicable to federated learning, arXiv preprint arXiv:2009.01974 (2020).

[16] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, C.-L. Li, Fixmatch: Simplifying semi-supervised learning with consistency and confidence, Advances in neural information processing systems 33 (2020) 596–608.

[17] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, T. Shinozaki, Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling, Advances in Neural Information Processing Systems 34 (2021) 18408–18419.

[18] Y. Wang, H. Chen, Q. Heng, W. Hou, Y. Fan, Z. Wu, J. Wang, M. Savvides, T. Shinozaki, B. Raj, et al., Freematch: Self-adaptive thresholding for semi-supervised learning, arXiv preprint arXiv:2205.07246 (2022).

[19] H. Chen, R. Tao, Y. Fan, Y. Wang, J. Wang, B. Schiele, X. Xie, B. Raj, M. Savvides, Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning, arXiv preprint arXiv:2301.10921 (2023).

[20] M. Jiang, H. Yang, X. Li, Q. Liu, P.-A. Heng, Q. Dou, Dynamic bank learning for semi-supervised federated image diagnosis with class imbalance, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2022, pp. 196–206.

[21] S. Bai, S. Li, W. Zhuang, J. Zhang, K. Yang, J. Hou, S. Yi, S. Zhang, J. Gao, Combating data imbalances in federated semi-supervised learning with dual regulators, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 10989–10997.

[22] C. Fan, J. Hu, J. Huang, Private semi-supervised federated learning., in: IJCAI, 2022, pp. 2009–2015.

[23] D. Yang, Z. Xu, W. Li, A. Myronenko, H. R. Roth, S. Harmon, S. Xu, B. Turkbey, E. Turkbey, X. Wang, et al., Federated semi-supervised learning for covid region segmentation in chest ct using multi-national data from china, italy, japan, Medical image analysis 70 (2021) 101992.

[24] S. Zhu, X. Ma, G. Sun, Two-stage sampling with predicted distribution changes in federated semi-supervised learning, Knowledge-Based Systems 295 (2024) 111822.

[25] F.-A. Croitoru, V. Hondru, R. T. Ionescu, M. Shah, Diffusion models in vision: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (9) (2023) 10850–10869.

[26] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in: International conference on machine learning, PMLR, 2015, pp. 2256–2265.

[27] Y. Song, S. Ermon, Generative modeling by estimating gradients of the data distribution, Advances in neural information processing systems 32 (2019).

[28] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, B. Poole, Score-based generative modeling through stochastic differential equations, arXiv preprint arXiv:2011.13456 (2020).

[29] Z. You, Y. Zhong, F. Bao, J. Sun, C. Li, J. Zhu, Diffusion models and semi-supervised learners benefit mutually with few labels, Advances in Neural Information Processing Systems 36 (2024).

[30] M. Yang, S. Su, B. Li, X. Xue, Exploring one-shot semi-supervised federated learning with pre-trained diffusion models, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 16325–16333.

[31] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, T. Aila, Training generative adversarial networks with limited data, Advances in neural information processing systems 33 (2020) 12104–12114.

[32] B. Zhao, H. Bilen, Dataset condensation with differentiable siamese augmentation, in: International Conference on Machine Learning, PMLR, 2021, pp. 12674–12685.

[33] P. G. Mulloy, Smoothing data with faster moving averages, Stocks & Commodities 12 (1) (1994) 11–19.

[34] P. Kar, V. Chudasama, N. Onoe, P. Wasnik, Revisiting class imbalance for end-to-end semi-supervised object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 4570–4579.

[35] Q. Xie, Z. Dai, E. Hovy, T. Luong, Q. Le, Unsupervised data augmentation for consistency training, Advances in neural information processing systems 33 (2020) 6256–6268.

[36] Y. Kim, J. Yim, J. Yun, J. Kim, Nlnl: Negative learning for noisy labels, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 101–110.

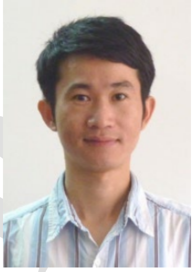
[37] Y. Chen, X. Tan, B. Zhao, Z. Chen, R. Song, J. Liang, X. Lu, Boosting semi-supervised learning by exploiting all unlabeled data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 7548–7557.

Zeyuan Wang received his M.S. degree in Computer Technology from School of Computer and Electronic Information, Guangxi University, China. He received the B.S. degree in Information Security from Guangxi University in 2022. His current research interests include federated learning and semi-supervised learning.

Yang Liu obtained his D.Sc. degree in computer science and technology from the South China University of Technology, Guangzhou, China, in 2021. Presently, he serves as a Postdoctoral Research Fellow at the Center for Machine Vision and Signal Analysis, University of Oulu, Oulu, Finland. Additionally, he has contributed as a guest associate editor for Frontiers in Psychology and Frontiers in Human Neurosciences. His research focuses on multimodal affective computing, detection of digital face forgery, and analysis of animal motion.

Guirong Liang is a master's student in computer science and technology at Guangxi University. He received his B.E. degree in Software engineering from Guangxi University of Science and Technology in 2022. His current research interests include federated learning and semi-supervised learning.

Cheng Zhong, Ph.D., Master's tutor of Computer Science and Technology, Guangxi University, Ph.D. tutor of Bioinformatics, Guangxi University, Master's tutor of Electronic Information (Computer Technology), South China University of Technology; part-time Ph.D. tutor of Computer Science, South China University of Technology. His main research interests are parallel and distributed computing, high performance computing for biomedical information, data security and sensitive information protection.



Feng Yang is currently an associate professor at the School of Computer and Electronics and Information, Guangxi University, Nanning, Guangxi, China. He received the Ph.D. degree in biomedical engineering from Huazhong University of Science and Technology, Wuhan, China, in 2016. His research is mainly at interdisciplinary field of medical image analysis, parallel computing and artificial intelligence, such as computer aided diagnosis, federated learning, medical image segmentation, registration, and fusion.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: